Project #10: Adapting the Pre-trained Deep Model to Novel Categories Classification

TA: Zhou Linjun & Shen Zheyan

July 7, 2019

1 Introduction

In this experiment, you will use a pre-trained alexnet model based on ILSVRC2015 dataset, which contains the whole 1,000 classes image data to conduct a novel categories classification task. The new task mainly uses 50 classes of the Caltech256 dataset. You will encounter different shots (1/5/20/50) in the training data and test 3 different algorithms on the same testing data.

2 Let's Get Started

We recommend you to complete the following 5 tasks in 150 minutes.

2.1 AlexNet

Read the file *src/alexnet.py*, and get a brief overview about the implement of the AlexNet. You should describe the structure and some details of the AlexNet in your own words in your report.

Hint: First, see function *create()*. This is a summary of the AlexNet and we recommend you to describe it briefly in your report. Then, see function *conv()* and function *max_pool()*. For convolutional layers, we recommend you explain the following four concepts in your report: Convolutional Filters, Stride, Max Pooling and Padding. For fully-connected layers, we recommend you explain the concept RELU in your report. After reading these functions, thinking how these concepts are used in the whole network. Note that there's another function *lrn()*, which is called Local Response Normalization. You don't need to know how it works. What you just need to know that it is a normalization method and could speed up the training process.

2.2 Softmax Regression

Read the file *src/finetune.py*, and get acquainted with the procedure of the novel categories classification. However, the file lacks some core codes. You should complete the whole file by filling in some codes.

Hint: You should first analyze the structure of the whole project, which we also give in Part 3. Then read file *src/finetune.py* carefully. There will be an annotation in the file to tell you where and how to add your code.

2.3 Fine-tuning

After you finish 2.2, you have completed tuning parameter W_{78} while keeping other parameters unchanged. In this sub-task, for simplicity you should tune both W_{78} and W_{67} with different learning rate: 0.01 and 0.001 while keeping other parameters unchanged.

Hint: You should learn how to adapt different learning rate for different parameters in Tensorflow. You could also find the hint in the file *src/finetune.py*

2.4 Cosine Similarity Metric Based Method (Bonus)

In this task, you will learn how to extract fc7 layer feature for each image and apply the metric based method to novel categories classification.

Hint: First, you should get a copy of your file *src/finetune.py* after finishing 2.3 and adjust some code on your copy file. You should try to get fc7 feature for both training data and testing data. *numpy.save()* and *numpy.load()* will be two good functions to store the features. Then, you could complete the whole algorithm in a new file.

2.5 Comparison of the Three Methods (Bonus)

You should complete the experiment result table shown in the ppt file. Totally you will do 12 experiments, including 3 different methods using on 4 different training data (1 shot / 5 shots / 20 shots / 50 shots). Then give some interesting findings in your report.

3 Project Directory Structure

- data/bvlc_alexnet.npy: Pre-trained AlexNet Model Parameters
- data/caltech_train_*.txt: Training data for novel categories (totally 4 groups)
- data/caltech_test.txt: Testing data for novel categories
- dataset/: Whole Caltech256 images we use in the experiment
- record/: Record for snapshot of the training process. You may not use it in the experiment.
- src/alexnet.py: The structure of the AlexNet.
- src/datagenerator.py: Convert the image to original vector used for training or testing. You don't need to know the implement of the file. Just use it.
- src/finetune.py: Main entry of the project. However it now lacks some core code. You need to complete the file.

4 Requirements

After you complete all tasks, you should include the followings in your project directory:

- A fixing finetune.py
- A file used to extract fc7 feature of the training data and testing data based on finetune.py
- A file including the algorithm of the task 2.4
- A report including answering the question of task 2.1 and the final result table of task 2.5

The experiment is relatively difficult. So please feel free to ask for the help of the TAs if you encounter any problem.